

REMARKS

Claims 1, 2, 8, 9, 13, and 17 are amended. Claim 3 was previously canceled without prejudice or disclaimer. Claims 1-2 and 4-20 are pending. No new matter is added by these amendments. By amending the claims, applicants are not conceding that the claims are unstatutory under 35 U.S.C. 101 and 112, as the present claim amendments are only for the purpose of facilitating expeditious prosecution. Applicant respectfully reserves the right to pursue these and other claims in one or more continuations and/or divisional applications. Applicant respectfully requests reconsideration and allowance of all claims in view of the amendments above and the remarks that follow.

Claim Rejections under 35 U.S.C. 101 and 112

Claims 1, 2, and 4-20 are rejected under 35 U.S.C. 101 because “the claimed invention is directed to an invention that is not useful. The disclosed invention is inoperative and therefore lacks utility. Examiner does not understand how this invention works.” Claims 1, 2, and 4-20 are rejected under 35 U.S.C. 112 because “the claimed invention is not supported by either a credible asserted utility or a well established utility for the reasons set forth above, one skilled in the art clearly would not know how to use the claimed invention.” Applicant respectfully traverses these grounds for rejection for the reasons argued below.

Applicant’s specification at page 2 lines 5-27 recites:

“A good programmer knows that the way to avoid problems caused by the copy-and-paste technique is to create a separate routine that includes the code and call the routine from both the first and the second location. Thus, the code only exists in one place: the separate routine. If the code ever requires modifications in the future, the programmer need only modify the separate routine, and the programmer is not burdened with trying to remember all of the locations where

the code exists. Unfortunately, programmers rarely take the time and effort to create a separate routine because it requires additional work at the moment, and programmers are routinely rushed and under pressure to work quickly. Without a better way to handle code modifications, computer software will continue to suffer from errors and fragility, which results in less reliable computer software, which is more expensive to maintain. ... In this way, future modifications to the block of code need only be made in one place: the method, which may reduce the fragility of the code.”

Thus, claims 1-2 and 4-20 have utility because the claimed invention creates a method where “future modifications to the block of code need only be made in one place: the method, which may reduce the fragility of the code,” as recited in applicant’s specification at page 2, lines 25-27.

Applicant respectfully submits that the Office Action did not establish a prima facie showing of lack of utility under 35 U.S.C. 101 or 112. As recited in MPEP 2107(II)(C), a prima facie showing of lack of utility under must include: a detailed explanation why the claimed invention has no specific and substantial credible utility, including documentary evidence to support the factual basis for the prima facie showing of no specific and substantial credible utility. If documentary evidence is not available, then the prima facie showing must specifically explain the scientific basis for the factual conclusions.

(1) Where the asserted utility is not specific or substantial, a prima facie showing must establish that it is more likely than not that a person of ordinary skill in the art would not consider that any utility asserted by the applicant would be specific and substantial. The prima facie showing must contain the following elements:

(i) An explanation that clearly sets forth the reasoning used in concluding that the asserted utility for the claimed invention is not both specific and substantial nor well-established;

(ii) Support for factual findings relied upon in reaching this conclusion; and

(iii) An evaluation of all relevant evidence of record, including utilities taught in the closest prior art.

(2) Where the asserted specific and substantial utility is not credible, a prima facie showing of no specific and substantial credible utility must establish that it is more likely than not that a person skilled in the art would not consider credible any specific and substantial utility asserted by the applicant for the claimed invention. The prima facie showing must contain the following elements:

(i) An explanation that clearly sets forth the reasoning used in concluding that the asserted specific and substantial utility is not credible;

(ii) Support for factual findings relied upon in reaching this conclusion; and

(iii) An evaluation of all relevant evidence of record, including utilities taught in the closest prior art.

Further, the prima facie showing must treat as true a statement of fact made by an applicant in relation to an asserted utility, unless countervailing evidence can be provided that shows that one of ordinary skill in the art would have a legitimate basis to doubt the credibility of such a statement.

Applicant respectfully submits that the reasoning in the Office Action “the claimed invention is directed to an invention that is not useful. The disclosed invention is inoperative and therefore lacks utility. Examiner does not understand how this invention works” and “the claimed invention is not supported by either a credible asserted utility or a well established utility for the reasons set forth above, one skilled in the art clearly would not know how to use the claimed invention” is merely conclusory, does not provide reasoning, does not provide support for factual findings, does not provide an evaluation of all relevant evidence of record, including utilities taught in the closest prior art, and does not establish that it is more likely than not that a person skilled in the art

would not consider credible the utility asserted at page 2, lines 5-27, of applicant's specification. Thus, applicant respectfully submits that the Office Action does not establish a prima facie case of a lack of utility.

Responses to the Examiner's questions:

Regarding claim 1:

1. "[P]resumably this references an IDE editor command that selects a block of code and a paste location."

Response: An IDE editor is one example embodiment of the invention, as recited at page 5, lines 4-18 of applicant's specification, by way of example and not of limitation.

2. "[T]he IDE editor creates a method and copies in the block of code?"

Response: One technique for creating a method that comprises a block of code is creating a method and then copying the block of code into the method, as recited in applicant's specification at page 15, lines 21-22 and page 16, lines 5-6, by way of example and not of limitation.

3. "At the paste location, insert an instruction that calls the newly created code method?"

Response: One example technique for adding an invocation of the method at the paste location is inserting an instruction that calls the method, by way of example and not of limitation.

4. "Where is the newly created code method located?"

Response: The location of the method is not the subject of claim 1. But, applicant's specification describes a variety of locations for the method, such as in a package, as recited on page 11, lines 4-18, by way of example and not of limitation.

5. "Is the code block 'cut' from its former location?"

Response: Cutting the code block is not the subject of claim 1. But, claims 8, 13, and 17 recite a “replacing” element, which could be implemented, in part, by cutting or removing the code block, by way of example and not of limitation.

6. “Is the newly created code method block inserted in the paste location?”

Response: No, claim 1 does not recite inserting the method at the paste location. Instead, claim 1 recites adding an invocation of the method at the paste location.

7. “Is a ‘goto’/call type of instruction inserted at the original code block location?”

Response: No, claim 1 does not recite adding anything at a location of the block of code. Instead, claim 1 recites adding an invocation of the method at the paste location. One example implementation of adding an invocation of the method is adding, at the paste location, an instruction that calls the method, by way of example and not of limitation.

Regarding claim 2:

8. “How did the block of code get to a ‘copy location’? Was it ‘cut’ from its original location and pasted into a copy location/temporary copy location?”

Response: No, the copy location is the original location of the block of code at the time of the receiving in claim 1.

9. “[T]he block of code ... is replaced with an invocation of the method? Is this a second invocation? This second invocation is at the copy location, while the first invocation is at the paste location?”

Response: Yes, by way of example and not of limitation.

10. Regarding claim 6: “Is the block of code ever ‘cut’ from its original locations or is it only copied into a new method block?”

Response: Cutting the block of code is not the subject of claim 6. But, claims 8, 13, and 17 recite a “replacing” element, which could be implemented, in part, by cutting or removing the code block, by way of example and not of limitation.

Regarding claim 7:

11. “Referencing the Specification, page 10, lines, does ‘punch-out’ mean a ‘cut and paste’ type of operation, with an invocation (method call) inserted at the location where the code block was cut?”

Response: A ‘punch-out’ operation is different from a cut and paste operation, but the punch-out operation does insert the invocation at the location where the block of code was cut, by way of example and not of limitation.

12. “The method call is invoking the newly created method and code block which has been inserted in its new location?”

Response: No, the invocation invokes the method, the invocation is at the paste location, and the method comprises the block of code, as recited in claim 1, on which claim 7 depends.

Regarding claim 8:

13. “An invocation/call to the newly created method, is inserted at the original location of the block of code?”

Response: Yes, by way of example and not of limitation.

14. “‘Replacing’ infers that the block code was ‘cut’?”

Response: Yes, by way of example and not of limitation.

15. “After the method comprising the block of code was created, where is it to be located? At the paste location?”

Response: No, the location of the method is not the subject of claim 8. But, applicant's specification describes a variety of locations for the method, such as in a package, as recited on page 11, lines 4-18, by way of example and not of limitation.

Regarding claim 9:

16. "Is this the first invocation? Is it already added at the original location of the block of code. A duplicate invocation is added at the paste location?"

Response: Claim 8 replaces the block of code with a first invocation of the method while claim 9 adds a second invocation of the method at the paste location, so there are two invocations of the method at different locations.

Regarding claim 13:

17. "The same invocation is inserted at the original location of the block of code and additionally at the paste location?"

Response: Yes, invocations of the method are added to two locations.

Conclusion

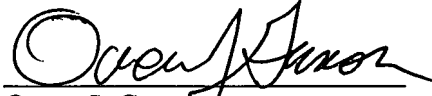
Applicant respectfully submits that the claims are in condition for allowance and notification to that effect is requested. The Examiner is invited to telephone applicant's attorney (651-645-7135) to facilitate prosecution of this application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 09-0465.

Respectfully submitted,

Eric L. Barnsness, et al.

By their Representative,



Date: May 29, 2007

Owen J. Gamon
Reg. No. 36,143
(651) 645-7135

IBM Corporation
Intellectual Property Law
Dept. 917, Bldg. 006-1
3605 Highway 52 North
Rochester, MN 55901

CERTIFICATE UNDER 37 CFR 1.8: I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on May 29, 2007.

Owen J. Gamon
Name



Signature